

---

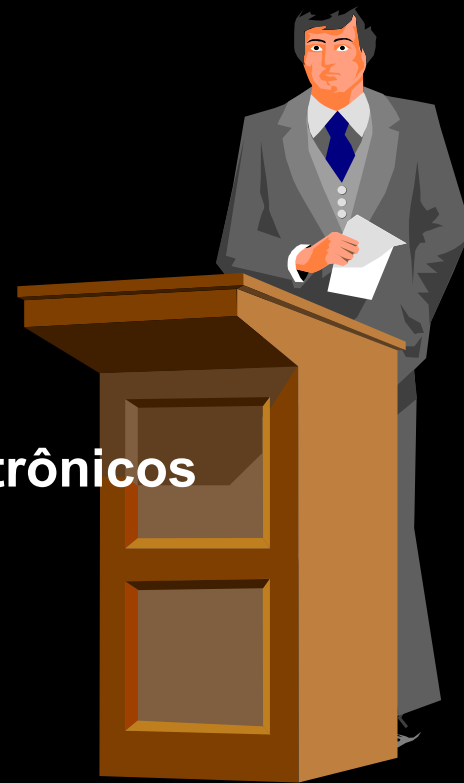
# **PSI 3541**

## **Sistemas Embarcados Distribuídos**

### **Aula 03**

**Prof. Dr. Sergio Takeo Kofuji**  
**Dr. Volnys Borges Bernal**

**Departamento de Engenharia de Sistemas Eletrônicos**  
**Escola Politécnica**  
**Universidade de São Paulo**



---

# REDES PARA APLICAÇÕES DE CONTROLE

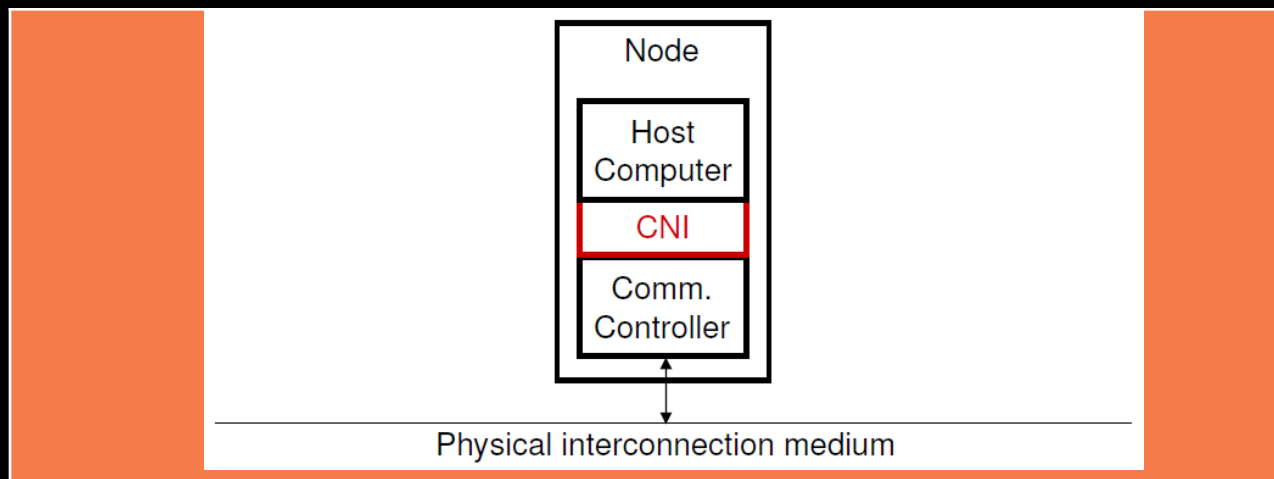
# Adaptar a Rede para Controle

---

- ❑ o subsistema de comunicação deve ser adaptado às características específicas das aplicações de controle.
- ❑ A seguir:
  - ❖ esclarecer o que é o subsistema de comunicação e qual sua função básica
  - ❖ alguns requisitos relacionados ao tráfego
  - ❖ adaptação do modelo OSI
  - ❖ Para cada camada do modelo OSI descrever algumas características típicas deste tipo de redes

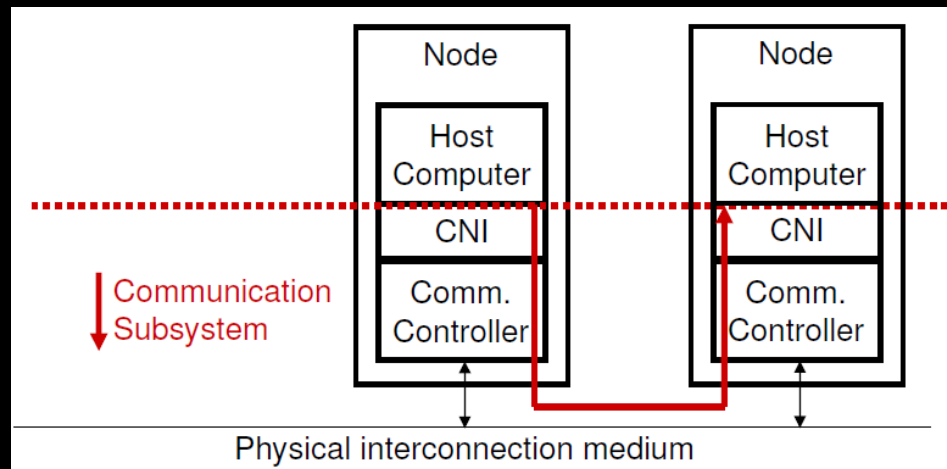
# O subsistema de comunicação

- ❑ Um nó geralmente inclui pelo menos dois componentes: o controlador de comunicação local e o computador host
  - ❖ A interface de rede de comunicação (CNI) entre host e comm. controlador está localizado no nível de transporte da pilha OSI



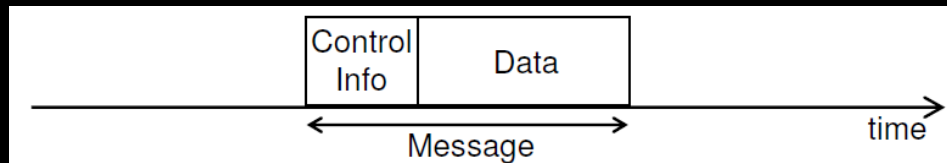
# O Subsistema de Comunicação

- ❑ O subsistema de comunicação inclui desde o meio de interconexão até as CNIs de cada nó
  - ❖ O objetivo deste sistema é transportar mensagens da CNI do remetente para a CNI do destinatário de forma confiável e de maneira temporizada (“timely”).



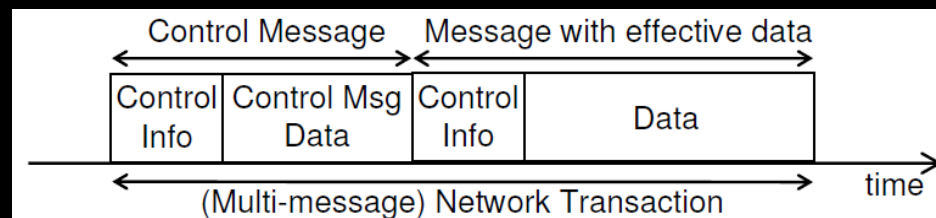
# O Subsistema de Comunicação

- ❑ Troca de mensagens é o mecanismo usado pelos nós para interagir entre eles
- ❑ Uma mensagem é uma unidade de informação que deve ser transferida, em um determinado momento, de um processo emissor para um ou mais processos receptores.
- ❑ Contém os respectivos dados, bem como as informações de controle que são relevantes para a transmissão adequada dos dados (por exemplo, remetente, destino, conteúdo).



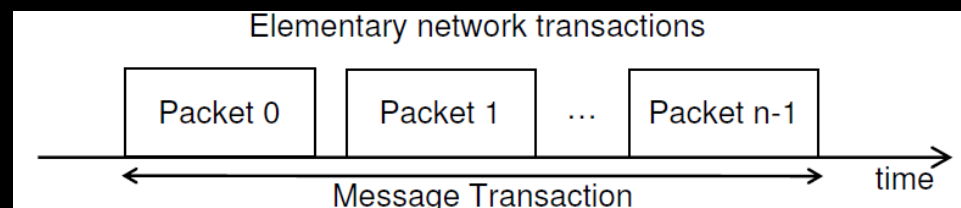
# O Subsistema de Comunicação

- ❑ Uma transação de rede é a sequência de ações dentro do sistema de comunicação necessária para realizar a transferência efetiva dos dados de uma mensagem.
- ❑ Uma transação pode incluir apenas a transferência de mensagens que transportam informações de controle de protocolo. Estes são referidos como mensagens de controle.
- ❑ Uma transação com várias mensagens é atômica quando todas as suas mensagens devem ser transmitidas sem interrupção.



# O Subsistema de Comunicação

- ❑ Muitas redes quebram automaticamente mensagens grandes em pacotes menores (fragmentação / remontagem).
- ❑ Nesse caso, uma transação de mensagem inclui várias transações de rede elementares que correspondem à transferência dos respectivos pacotes.
- ❑ Um pacote é a menor unidade de informação que é transmitida sem interrupção (quando não há risco de confusão, usaremos mensagens e pacotes de forma intercambiável).





# O Subsistema de Comunicação

---

- ❑ A eficiência de dados do protocolo de rede pode ser definida como a razão entre o tempo para transmitir bits de dados efetivos e a duração total da transação respectiva.
- ❑  $\text{Data\_eff} = \text{data\_tx\_time} / \text{transaction\_duration}$
- ❑ Em geral: quanto mais curtos os dados por transação, menor é a eficiência

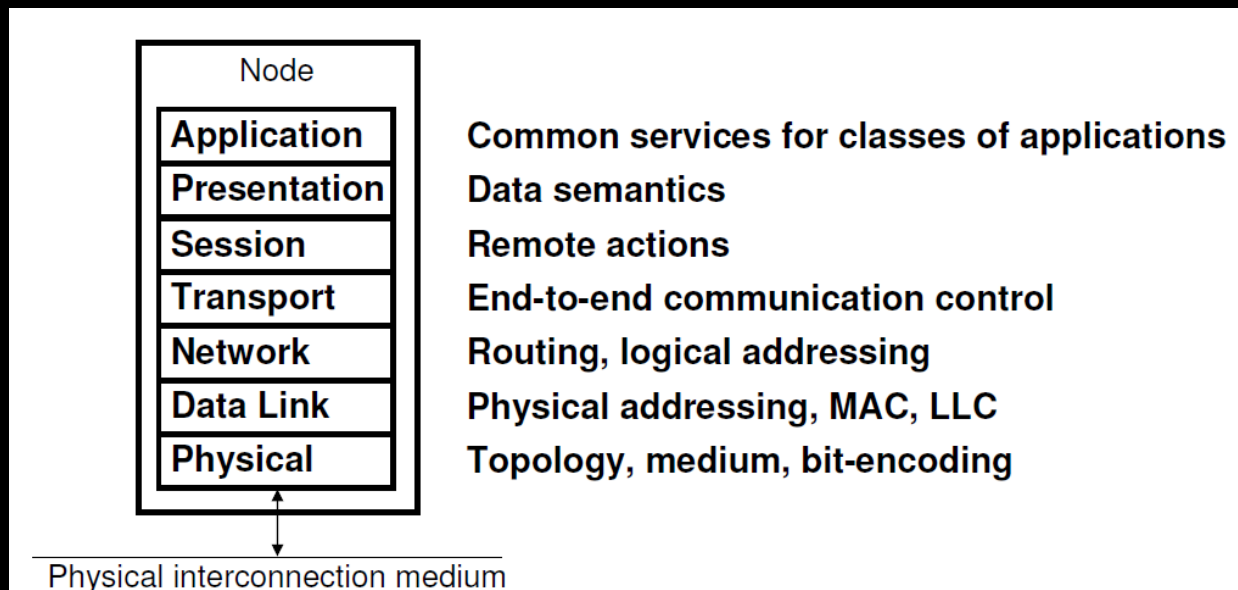
# Requisitos relacionados ao tráfego

---

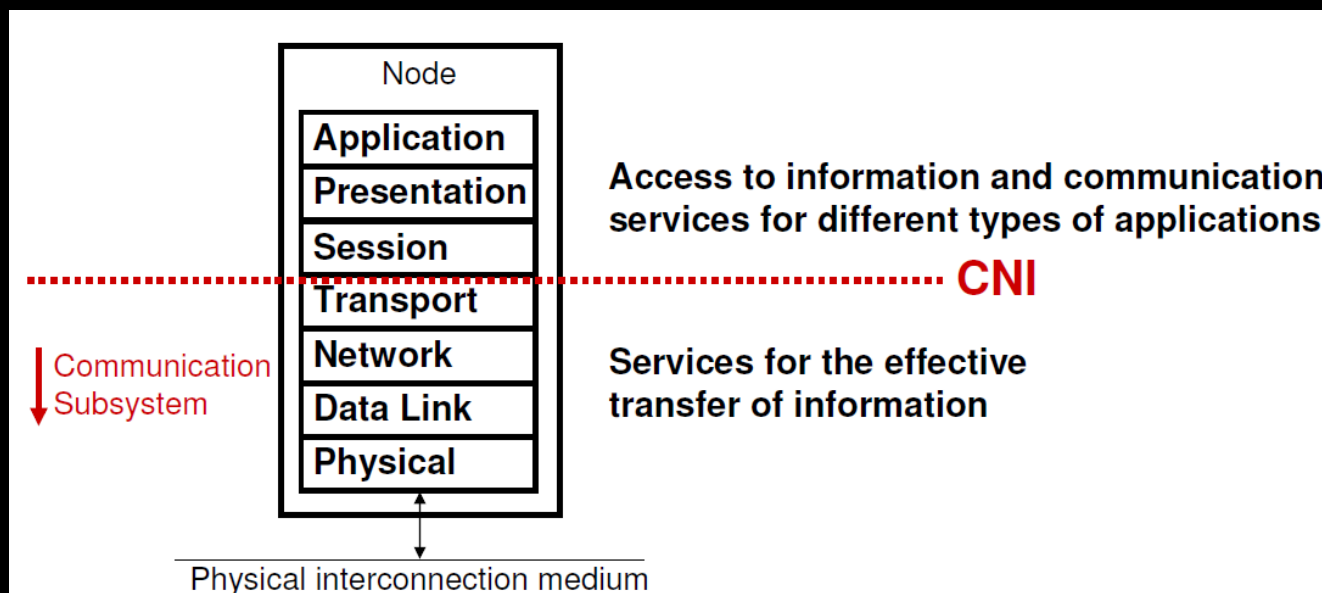
- ❑ **Transmissão eficiente de dados curtos (poucos bytes)**
- ❑ **Transmissão periódica (monitoramento, controle de feedback) com períodos curtos (ms) e baixa instabilidade (variação de tempo)**
- ❑ **Transmissão rápida (ms) de solicitações aperiódicas (alarmes)**
- ❑ **Transmissão de dados não tempo real (configuração, logs)**
- ❑ **Multicasting**

# Modelo OSI

- **A pilha OSI (modelo de referência para sistemas abertos)**
  - ❖ Cada camada fornece serviços específicos para o acima
  - ❖ Cada camada adiciona algumas informações de controle à mensagem



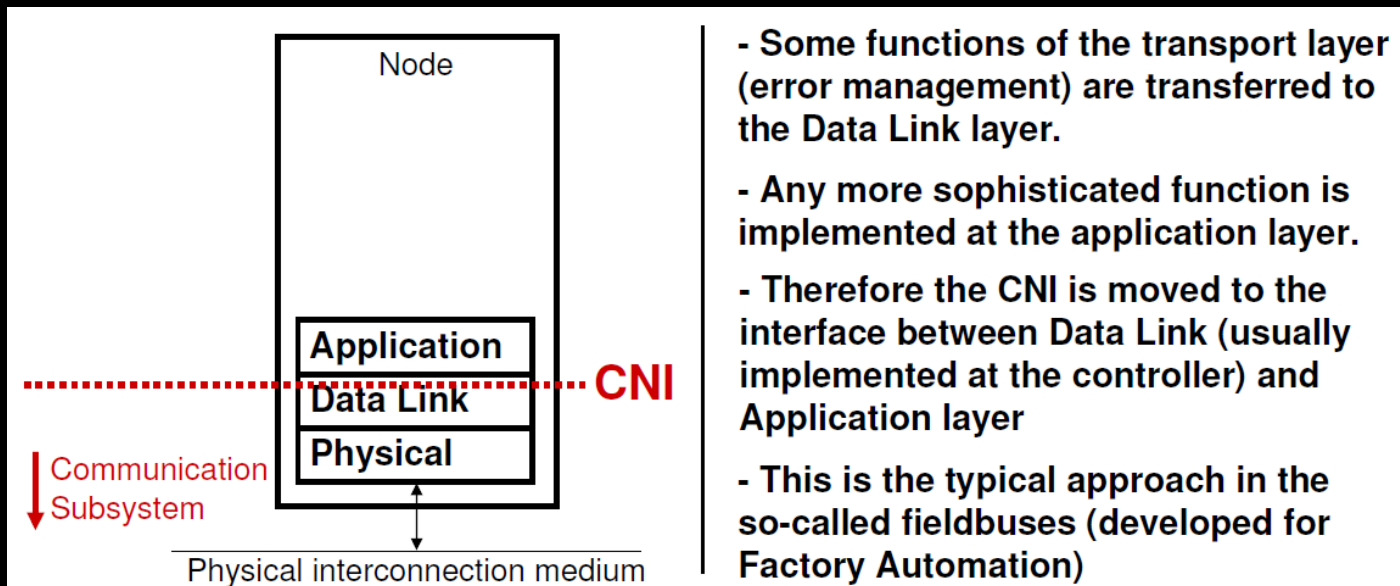
# Pilha OSI e CNI



# OSI simplificado (colapsado)

## ❑ O modelo recolhido do OSI é usado

- ❖ As sete camadas são recolhidas em três: Físico, Data Link e Application.



# Camada Física

---

- ❑ **Em muitos casos, o baixo custo do esquema de cabeamento é fundamental!**
- ❑ **Portanto, a topologia de barramento tem sido a favorita.**
  - ❖ Simplifica o cabeamento
  - ❖ Torna desnecessário para o protocolo executar o roteamento
  - ❖ Facilita uma recepção consistente da mesma informação no modo de transmissão,
  - ❖ Facilita a obtenção de uma ordem consistente na recepção das mensagens.
- ❑ **Algumas outras topologias são adequadas para objetivos específicos**

# Camada de Enlace de Dados

---

- ❑ **Entre as funções da Camada de Enlace de Dados, discutiremos brevemente alguns aspectos de:**
  - ❖ Endereçamento
  - ❖ Controle de Link Lógico (LLC)
  - ❖ Controle de Acesso ao Meio (MAC)
    - Fundamental para a resposta em tempo real RT

# Camada de Enlace de Dados

---

- **Endereçamento: identificação das partes envolvidas em uma transação de rede**
  - ❖ Endereçamento direto: o remetente e o destinatário (s) são explicitamente identificados em cada transação, usando endereços físicos (como na Ethernet)
  - ❖ Endereçamento indireto (fonte): O conteúdo da mensagem é explicitamente identificado (por exemplo, temperatura do sensor X). Receptores que precisam da mensagem, recuperam da rede (como no CAN e no WorldFIP)
  - ❖ Endereçamento indireto (baseado em tempo): A mensagem é identificada pelo instante de tempo no qual ela é transmitida (como no TTP)



- 
- ❑ **Controle de Link Lógico (LLC). Ex. controle de erros de transmissão**
  - ❑ **Erros devem ser detectados e, em seguida, alguma ação deve ser tomada para resolver o problema:**
    - ❖ Correção de erro antecipada (FEC)
      - Códigos de Correção de erro (mais relacionados à camada física)
      - Ou o receptor aguarda a próxima transmissão periódica
    - ❖ Solicitação de Repetição Automática (ARQ) O receptor aciona um pedido de repetição mediante erro
    - ❖ Reconhecimento positivo e repetição (PAR) O remetente reenvia se o ACK não for recebido
    - ❖ De uma perspectiva em tempo real, ARQ e PAR podem induzir atrasos de entrega mais longos e carga de comunicação extra

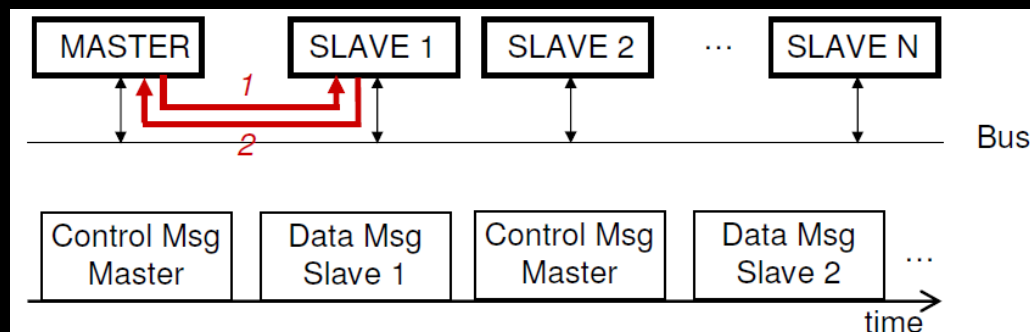
---

## □ MAC

- ❖ Decide quem deve obter acesso ao meio compartilhado (se houver algum).
- ❖ Portanto, determina o atraso de acesso à rede
- ❖ É fundamental para o comportamento em tempo real de uma rede
- ❖ Exemplos:
  - Master/Slave
  - TDMA
  - CSMA / CD
  - CSMA / BA
  - CSMA / CA

## ❑ Controle de Acesso Médio (MAC): Master-Slave

- ❖ Acesso concedido pelo mestre
- ❖ Msgs escravo podem ser endereçadas a qualquer outro nó
- ❖ “Natural” em muitas aplicações de automação. O controlador central se torna o mestre das comunicações
- ❖ Nós sincronizados com o mestre
- ❖ Requer uma mensagem de controle por mensagem de dados



---

## ❑ **Controle de Acesso Médio (MAC): Master-Slave**

- ❖ O problema de programação de tráfego se torna local para o mestre. Isto implica em uma boa flexibilidade em relação a algoritmos de escalonamento (on-line ou off-line, qualquer tipo de agendamento de processador)
- ❖ O mestre representa um único ponto de falha. Para alta confiabilidade, o mestre deve ser replicado
  - Observe que, em sistemas de controle hierárquico, o mestre provavelmente é o controlador “principal”. Portanto, já é um ponto único de falha em relação ao controle.

## ❑ **Mensagens mestras são suportes naturais de pontos de sincronização trigger tx preciso**

- ❖ Observe que os instantes de recepção podem variar de nó para nó, dependendo da distância até o mestre

## ❑ **Exemplos: WorldFIP ou Ethernet Powerlink**

# exercício

---

## □ CSMA

### ❖ CSMA-CD

- Shared ethernet (HUB)

### ❖ CSMA-BA

- CAN

### ❖ CSMA-CA

- FlexRay-async (Byteflight), ARINC629-async



# Camada de Aplicação

---

- **Entre as funções relacionadas à Camada de Aplicação, vamos revisar alguns aspectos de:**
  - ❖ Modelos de cooperação
  - ❖ MMS
  - ❖ Sincronização de relógios

# Cont...

---

## □ **Modelo de cooperação: cliente-servidor**

- ❖ As transações são acionadas pelo receptor da informação solicitada (cliente).
- ❖ Os nós que geram informações são servidores e só reagem às solicitações do cliente.
- ❖ O modelo é baseado na transmissão unicast (um remetente e um receptor)



# Cont..

---

- ❑ **Modelo de cooperação: produtor-consumidor**
  - ❖ As transações são acionadas pelos nós que geram informação (produtores).
  - ❖ Os nós que precisam da informação, identificam-na quando transmitida e a recuperam da rede (consumidores)
  - ❖ O modelo é baseado na transmissão de difusão (cada mensagem é recebida por todos os nós)

- 
- ❑ **Modelo de cooperação: produtor-distribuidor-consumidor**
  - ❑ **Basicamente semelhante ao Produtor-Consumidor**
  - ❑ **As transações são acionadas por um determinado nó, o distribuidor, mediante solicitação dos produtores ou de acordo com um cronograma (schedule) pré-estabelecido.**
  - ❑ **É uma implementação de Produtor-Consumidor (PC) sobre mestre-escravo**

- 
- ❑ **Modelo de cooperação: publiser-subscriber**
  - ❑ **Versão elaborada do produtor-consumidor usando o conceito de comunicação em grupo**
  - ❑ **Os nós devem aderir a grupos como publisher (produz informações) ou como subscriber (consome informações)**
  - ❑ **As transações são acionadas pelo publisher de um grupo e divulgadas entre os respectivos subscriber (multicast).**

---

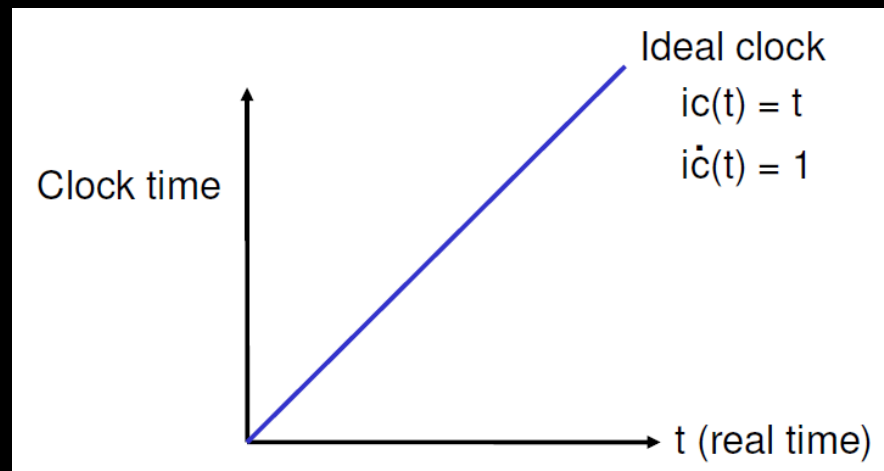
## ❑ **Manufacturing Message Specification (MMS)**

- ❖ OSI application layer messaging protocol
- ❖ Exchange of real-time data and supervisory control information between networked devices and applications
- ❖ Defines common functions for distributed automation systems
- ❖ Originally published in 1990 by ISO TC 184, (application layer of GM MAP)
- ❖ Standardized as ISO 9506-1/2
- ❖ Nowadays implemented in a wide range of networks (Ethernet, fieldbuses, RS485,

- 
- ❑ **Sincronização de Relógio - Necessária em Sistemas Distribuídos para ter uma noção comum de tempo para:**
  - ❑ **Realizar ações nos instantes de tempo desejados**
    - ❖ ex. aquisição de dados síncronos, atuação síncrona
  - ❑ **Time-stamp de Dados e eventos**
    - ❖ Ex. estabelecer relações causais que levaram a uma falha no sistema
  - ❑ **Calcular a idade dos dados**
  - ❑ **Sincronização de Comunicações**
    - ❖ ex. Esquemas de comunicação TDMA
  - ❑ **De fato, vários mecanismos para melhorar a confiabilidade dependem da presença de tal serviço. Ex.**
    - ❖ Contenção de erros no domínio do tempo
    - ❖ Esquemas de replicação de canal e nó
  - ❑ **Mas o sincronismo do relógio não é obrigatório**

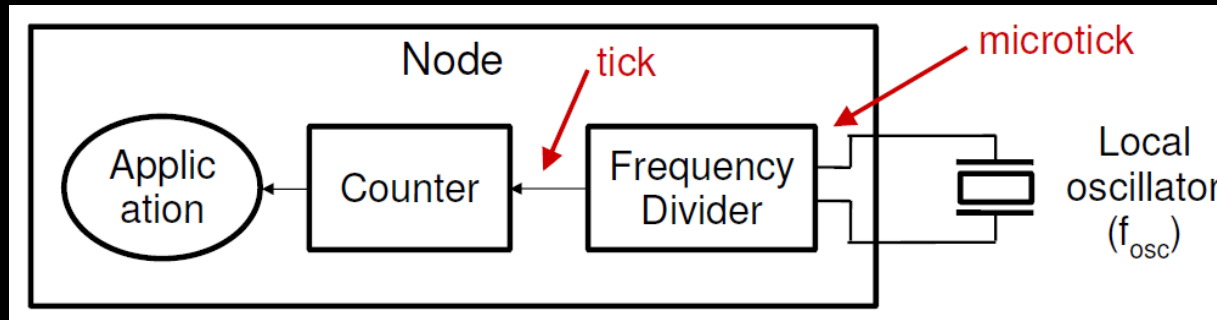
- 
- ❑ **Clock Synch - Alguns conceitos relacionados ao tempo:**
  - ❑ **Newtonian Time (o modelo utilizado em sistemas computacionais): o tempo é considerado uma dimensão externa e contínua, que é percebida igualmente em todos os lugares.**
  - ❑ **Esta noção de tempo, muitas vezes referida como tempo real, pode ser representada pelo conjunto de números reais positivos.**
  - ❑ **Qualquer instante que pertença a esta dimensão externa é geralmente denotado com a letra  $t$ .**

- ❑ **Clock Synch - Alguns conceitos relacionados ao tempo:**
- ❑ **- Relógio Ideal: o relógio que sempre reflete o valor do tempo real. Este relógio não tem existência física e pode ser considerado apenas teoricamente. AKA: relógio newtoniano, relógio perfeito**



## ❑ Clock Synch - Alguns conceitos relacionados ao tempo:

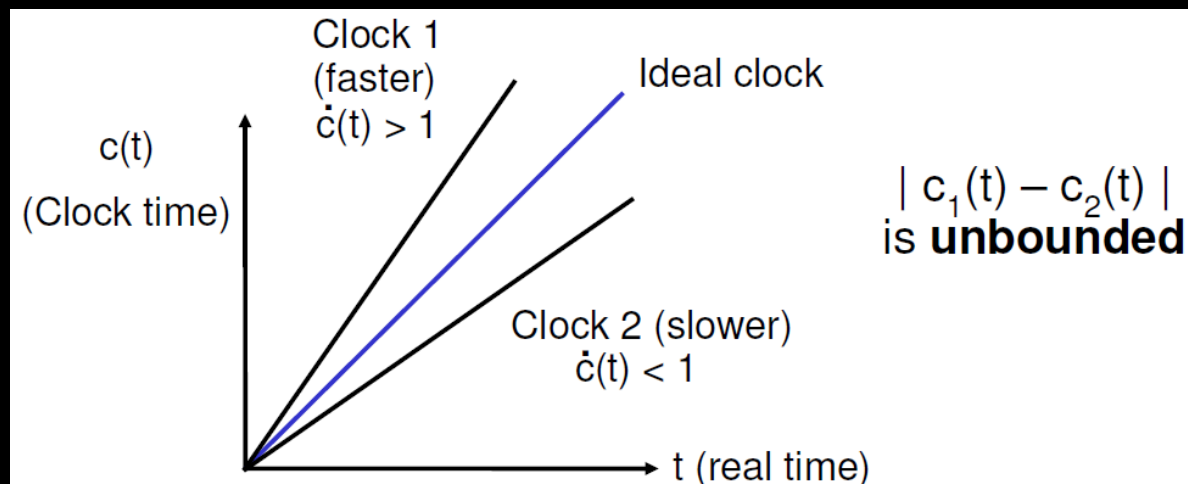
- ❖ Physical Clock: o valor de um contador que é incrementado de acordo com um oscilador local em cada nó. Este relógio tem existência física. AKA: relógio de hardware





## □ Clock Synch - Alguns conceitos relacionados ao tempo:

- ❖ Relógio físico: os osciladores locais desviam-se inevitavelmente de sua freqüência nominal:  $c'(t) = 1 + r(t)$ , onde  $r(t)$  é chamado de desvio do fs. relógio. Existe sempre um  $\text{Max}(|r(t)|)$ , chamado max. deriva.



---

## □ Clock Synch - Alguns conceitos relacionados ao tempo:

- ❖ Relógio Global: Alguns sistemas embarcados distribuídos são construídos sob a suposição de que todos os nós têm acesso a uma referência de tempo que é percebida igualmente em todos os lugares dentro do sistema. O relógio que fornece o valor dessa referência de tempo absoluto é chamado de relógio global.
  - Não garante que todos os nós realmente percebam igualmente!
  - Não mapeia o tempo real perfeitamente, mas pode desviar-se do tempo real.
  - AKA: Relógio absoluto, relógio de referência, relógio de todo o sistema
- ❖ O relógio global pode ter existência física ou não
  - Existência física, por exemplo, em uma sincronização do relógio mestre-escravo (o relógio físico do mestre é o relógio global)
  - Não há existência física quando o relógio global é definido como uma função (função de convergência) dos valores de vários relógios físicos (por exemplo, média do valor de vários relógios físicos)

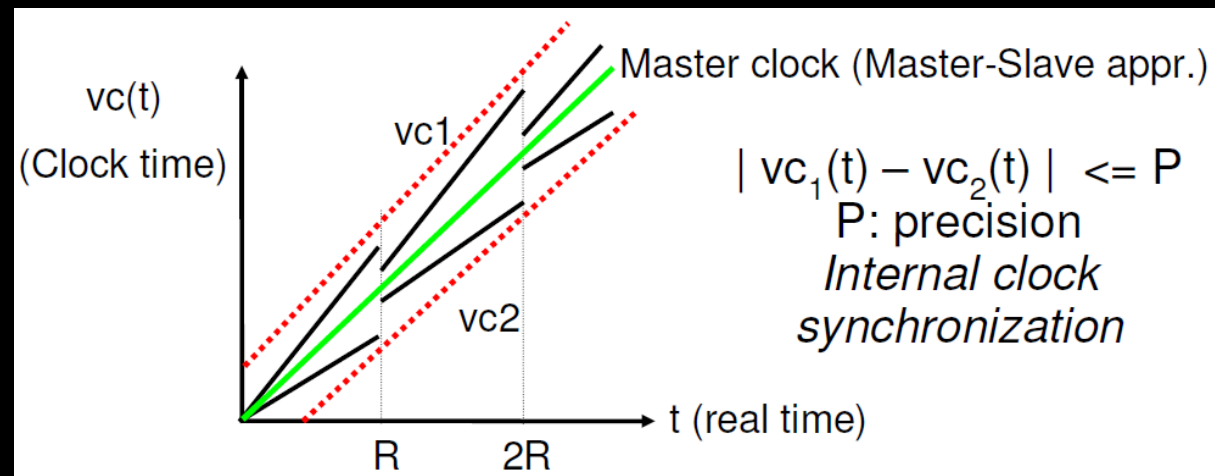
---

## □ **Clock Synch - Alguns conceitos relacionados ao tempo:**

- ❖ Relógio Virtual: É o resultado da sincronização de um relógio físico com o relógio global.
  - Os relógios físicos, por si só, não podem ser usados para aproximar o valor do relógio global, porque seus diferentes desvios os fazem divergir à medida que o tempo passa. Para resolver este problema, cada nó deve periodicamente executar um algoritmo de sincronização de relógio para estimar seu desvio em relação ao relógio global e corrigi-lo.
  - O relógio obtido após esta correção é aplicado no relógio físico e é chamado de relógio virtual, e é denotado como  $vc(t)$ .
  - Este relógio tem existência física: é o relógio físico quando é periodicamente sincronizado.
  - AKA: Relógio lógico, relógio sincronizado

## □ Clock Synch - Alguns conceitos relacionados ao tempo:

- ❖ Relógio virtual. Os três passos de um algoritmo de sincronização de relógio.
  - Detecção do instante de sincronização periódica.
  - Estimativa do valor do relógio global nesse instante de tempo e, portanto, do erro entre o relógio virtual e o relógio global.
  - Correção desse erro.



- 
- **Clock Synch - Mas este serviço nem sempre é necessário. Dois tipos de sistemas (paradigmas):**
    - ❖ Sistema acionado por temporizador: pressupõe que o relógio de cada nó é independente dos clocks dos demais nós (somente relógios físicos são usados). Cada nó só pode medir durações relativas ao seu próprio relógio físico
      - timestamps não fornecem nenhuma informação sobre a ordem dos eventos se tais eventos aconteceram em nós diferentes (tomados com diferentes relógios físicos).
      - Além disso, não é possível medir a duração das ações que começam em um nó e terminam em outro, ou seja, ações que são distribuídas.
      - Para conseguir alguma coordenação entre os nós, são usados temporizadores locais que são definidos pelos nós na ocorrência de um dado evento (por exemplo, a transmissão ou recepção de certas mensagens).
    - ❖ Sistema acionado pelo relógio: usa relógios virtuais (sincronizados).

---

## ❑ Sincronização do Relógio - Requisitos de Sincronização:

### ❖ Redes de dados genéricos:

- Aplicações: sistemas de arquivos distribuídos, transações financeiras, aplicativos de escritório
- Precisão: de milissegundos para segundos
- Protocolos: Network Time Protocol (NTP) (abrange a área de LAN e WAN)

### ❖ Sistemas distribuídos em tempo real:

- Aplicações: sistemas de supervisão, medição e controle
- Precisão: de sub-microsegundos para milissegundos
- Baixo custo! (dispositivos de baixo custo, poucos recursos de rede)
- Protocolos: IEEE1588, SynUTC, soluções adaptadas ao protocolo

---

## □ Sincronização do Relógio - IEEE1588

- ❖ Hierárquico, mestre / escravo
  - relógio grandmaster: melhor relógio no sistema
  - subnet master: melhor relógio em uma sub-rede  
(subnet única: grandmaster e master são os mesmos)
- ❖ Em cada sub-rede, os nós são sincronizados com o mestre de sub-rede
- ❖ Mestres de sub-rede sincronizam com o grande mestre
- ❖ A eleição do mestre é automática (algoritmo Best Master Clock).  
Mecanismos para indicação de um conjunto preferido de mestres
- ❖ Sincronização externa possível (por exemplo, GPS)

---

# **Comparative Study of Clock Synchronization Algorithms in Distributed Systems**

**Neha N. Dalwadi<sup>1</sup> and Dr. Mamta C. Padole<sup>2</sup>**

*<sup>12</sup>The Maharaja Sayajirao University, Vadodara, Gujarat, India*



**Table 1.** Comparative Study of Synchronization Algorithms

Name of Parameter & Name of Algorithm	Type of Algorithm	Approach	Scalability	Reasons for Implementation	Fault Tolerance	Limitation
<b>Cristian's Algorithm</b>	Centralized	Passive Time Server and based on External clock synchronization approach	Poor	To minimize propagation time (in milliseconds)	Not	1. Single time server might be fail. 2. Faulty Time server cause server replied with incorrect time.

<b>Berkeley Algorithm</b>	Centralized	Active Time Server and based on Internal clock synchronization approach	Poor	To minimize the maximum difference between any two clock (in milliseconds)	Not	1. Server becomes bottleneck.
<b>Global Averaging Algorithm</b>	Distributed	No such time server and based on internal clock	Poor	To resolve single point of failure and to minimize skew value (in milliseconds)	Not	1. Network should support broadcast facility. 2. Congestion may occur due to large amount of message passing.
<b>Network Time Protocol</b>	Distributed	External Clock is used as reference time server. Based on Multiple time server arranged in levels.	Good	To minimize propagation time (in milliseconds) and faster access of correct time value.	Yes	1. Supports in UNIX system only
<b>Precision Time Protocol</b>	Centralized	Master-Slave approach where Master is controlled by GPS receiver	Good	More accuracy than NTP by using GPS receiver, order of timing in (in microseconds)	Yes	1. Network should support multicasting. 2. Intended for relatively localized system.

---

Duvidas?

**KOFUJI@USP.BR**